

**Claims:**

We claim:

1. A process for transferring pen data between unmanaged and managed code comprising the step of:
  - receiving pen data in a component written in unmanaged code;
  - transferring information related to said pen data to a shared memory;
  - transferring a pointer that points to said information in said shared memory to an application written at least in part in managed code; and
  - retrieving said information from said shared memory.
2. The process according to claim 1, further comprising the steps of:
  - transferring additional information from said at least in part managed application to said shared memory;
  - transferring a pointer that points to said additional information to said component;
  - retrieving said additional information from said shared memory.
3. The process according to claim 1, further comprising the step of:
  - using a P-invoke style interface crossing between unmanaged code and managed code.
4. The process according to claim 1, further comprising the step of:
  - exchanging information through a COM interface.
5. The process according to claim 1, said component being a pen services component.
6. The process according to claim 1, said application including a pen input managed client.
7. The process according to claim 1, said component receiving input from at least one pen device driver.
8. A system for transferring information between unmanaged code and managed code comprising:
  - a shared memory
  - a component that receives pen data and transfers information relating to said pen data to said shared memory and transfers a pointer that points to said information to an application having managed code;

said application having managed code receives said pointer and obtains said information from said shared memory.

9. The system according to claim 8, said component exposing a COM interface.

10. The system according to claim 8, said application using a P-Invoke-style command.

11. The system according to claim 8, said component including a pen services component.

12. The system according to claim 8, further comprising:  
at least one pen device driver sending information to said component.

13. The system according to claim 8, further comprising:  
said application including a pen input managed client.

14. A computer-readable medium having a program stored thereon for transferring information related to ink between an unmanaged component and an application including managed code, said program comprising the steps of:

receiving pen data in a component written in unmanaged code;

transferring information related to said pen data to a shared memory;

transferring a pointer that points to said information in said shared memory to an application written at least in part in managed code; and

retrieving said information from said shared memory.

15. The computer-readable medium according to claim 14, said program further comprising the steps of:

transferring additional information from said at least in part managed application to said shared memory;

transferring a pointer that points to said additional information to said component;

retrieving said additional information from said shared memory.

16. The computer-readable medium according to claim 14, said program further comprising the step of:

using a P-invoke style interface crossing between unmanaged code and managed code.

17. The computer-readable medium according to claim 14, said program further comprising the step of:

exchanging information through a COM interface.

18. The computer-readable medium according to claim 14, said component being a pen services component.

19. The computer-readable medium according to claim 14, said application including a pen input managed client.

20. The computer-readable medium according to claim 14, said component receiving input from at least one pen device driver.